The Astrophysical Journal Supplement Series, 252:19 (14pp), 2021 February https://doi.org/10.3847/1538-4365/abcc67

© 2021. The American Astronomical Society. All rights reserved.

The Last Journey. I. An Extreme-scale Simulation on the Mira Supercomputer

Katrin Heitmann[1], Nicholas Frontiere[1,2], Esteban Rangel[1,3], Patricia Larsen[1], Adrian Pope[2], Imran Sultan[1], Thomas Uram[3], Salman Habib[1,2], Hal Finkel[3], Danila Korytov[1,4], Eve Kovacs[1], Silvio Rizzi[3], Joe Insley[3], and Janet Y. K. Knowles[3]

[1] High Energy Physics Division, Argonne National Laboratory, Lemont, IL 60439, USA
[2] Computational Science Division, Argonne National Laboratory, Lemont, IL 60439, USA
[3] Argonne Leadership Computing Facility, Argonne National Laboratory, Lemont, IL 60439, USA
[4] Department of Physics, University of Chicago, Chicago, IL 60637, USA

Received 2020 June 2; revised 2020 November 13; accepted 2020 November 18; published 2021 January 27

Abstract

The Last Journey is a large-volume, gravity-only, cosmological N-body simulation evolving more than 1.24 trillion particles in a periodic box with a side length of 5.025 Gpc. It was implemented using the HACC simulation and analysis framework on the BG/Q system Mira. The cosmological parameters are chosen to be consistent with the results from the Planck satellite. A range of analysis tools have been run in situ to enable a diverse set of science projects and, at the same time, keep the resulting data amount manageable. Analysis outputs have been generated starting at redshift $z \sim 10$ to allow for construction of synthetic galaxy catalogs using a semianalytic modeling approach in postprocessing. As part of our in situ analysis pipeline, we employ a new method for tracking halo substructures, introducing the concept of subhalo cores. The production of multiwavelength synthetic sky maps is facilitated by generating particle light cones in situ, also beginning at $z \sim 10$. We provide an overview of the simulation setup and generated data products; a first set of analysis results is presented. A subset of the data is publicly available.

Unified Astronomy Thesaurus concepts: Cosmology (343); Large-scale structure of the universe (902); N-body simulations (1083)

1. Introduction

Cosmology has undergone a sea change over the last three decades. Remarkable improvements in survey observations have changed many aspects of the field from being qualitative in nature, and even speculative, to the present era of "precision cosmology." The current cosmological standard model, Lambda cold dark matter (ΛCDM), successfully describes all observations using only a handful of parameters, each determined at an accuracy level of a few percent (for recent parameter constraints, see, e.g., Planck Collaboration et al. 2020). Although ΛCDM is phenomenologically very successful, its two key ingredients, dark matter and dark energy, remain mysterious. Additionally, the origin of primordial fluctuations requires further investigation.

In order to probe deeper into the unknown, major ground- and space-based surveys are being constructed and proposed. Among ground-based surveys, DESI (Aghamousa et al. 2016) recently achieved first light, and the Vera C. Rubin Observatory is readying to carry out LSST (Abell et al. 2009; Abate et al. 2012). From space, surveys such as WFIRST (Doré et al. 2019), Euclid (Laureijs et al. 2011), and SPHEREx (Doré et al. 2014) promise major advances in obtaining a host of cosmological measurements. With the advent of these surveys, the quality of the observations and their statistical completeness have led to an increased and intense focus on understanding systematic errors in all aspects of the inference chain that cosmology, as an observational science, must employ. Major modeling and simulation challenges have to be overcome to enable systematic and robust exploration of the scales covered by the observations.

The requirements for simulations that cover the size and depth of upcoming cosmological surveys are very demanding; only a handful of such simulations have been carried out so far. The volume required to both enable the creation of light cones out to high redshift and properly sample rare cluster-mass objects is approximately (4–5 Gpc)^3. In order to capture galaxies with luminosity $\sim 0.1 L^\star$, halo masses of order $10^{11} h^{-1}$ $M_\odot$ must be adequately resolved. This sets a requirement on the mass resolution of the simulation of $m_p \sim 2 \cdot 10^9 h^{-1}$ $M_\odot$ or better and, by itself, implies the need for simulations that evolve trillions of particles. Galaxies below the luminosity thresholds of such simulations need to be separately modeled, as described, for example, in Korytov et al. (2019).

For the Euclid survey, such a simulation was carried out with the tree code PKDGRAV3 on the Piz Daint supercomputer, a GPU-enhanced system. The Euclid Flagship simulation (Potter et al. 2017) evolved 12,600^3 particles in a (3780 h^{-1} Mpc)^3 volume with a mass resolution of $2.4 \cdot 10^9 h^{-1}$ $M_\odot$. Another simulation of similar size is the Outer Rim run (Heitmann et al. 2019b), carried out with the Hardware/Hybrid Accelerated Cosmology Code (HACC) on the IBM BG/Q system Mira in 2014. In that simulation, 10,240^3 particles in a (3000 h^{-1} Mpc)^3 volume were evolved, leading to a mass resolution of $1.85 \cdot 10^9$ $h^{-1} M_\odot$. The Outer Rim simulation (Heitmann et al. 2019b) has been used to generate synthetic skies for surveys such as LSST (Korytov et al. 2019), the South Pole Telescope (SPT; Bleem et al. 2020), and eBOSS (Avila et al. 2020; Alam et al. 2020; Gil-Marín et al. 2018; Hou et al. 2018; Rossi et al. 2020; Smith et al. 2020; Zarrouk et al. 2018). There are many other cosmological simulations available, occupying useful niches in the trade-off space of mass resolution and cosmological volume, but there are relatively few that simultaneously have adequate mass resolution and simulation volume to be successfully employed in modeling contemporary surveys.

In this paper, we describe the Last Journey simulation, a new addition to the set of large-scale structure simulations that are specifically targeted at survey science. As with the Outer Rim simulation, the Last Journey run was carried out on Mira, a supercomputer at the Argonne Leadership Computing Facility (ALCF). Mira, a 10 PFlop system, belongs to the family of

IBM BG/Q supercomputers and went into production in 2012. The machine was retired early in 2020, and the Last Journey simulation was its final full-machine run. The simulation was performed with HACC, described in detail in Habib et al. (2016). The HACC framework has run very successfully on the BG/Q architecture; the Gordon Bell finalist paper in 2012 (Habib et al. 2012) describes achieving a sustained performance of close to 14 PFlops on Sequoia, another large BG/Q system.

The Outer Rim simulation was one of the first extreme-scale simulations run on Mira, evolving more than a trillion particles and employing two-thirds of the machine. In this context, "extreme-scale" refers to those simulations that occupy a major fraction of a system that is among the largest supercomputers worldwide. Experience with the Outer Rim led to many optimizations in the HACC framework. These optimizations were applied to input/output (I/O) strategies, memory management, time stepping, and improved analysis tools. In the end, essentially every part of the code was touched and enhanced in some way. For the Last Journey run, we focused on a few additional improvements to the analysis suite. Given that Mira was on the floor for only a few more months with limited storage space availability, most of the analysis had to be carried out in situ, while the code was running on the system. This approach avoids having to store petabytes of data on the file system and also saves I/O time. In particular, we focused on speeding up the halo center finding algorithm, generating light-cone particle files in situ, and adding a range of halo properties to the output that were not measured for the Outer Rim simulation. We sought input from the LSST Dark Energy Science Collaboration (DESC) working groups and others to enable as many science projects related to major surveys as possible.

Over the years, we have developed a customized approach to the analysis that accompanies HACC simulations to ensure high performance and excellent scalability of our in situ tools. Our parallel analysis tool suite, called CosmoTools, takes advantage of the data layout of the main code and reuses optimized HACC algorithms, such as the large-scale distributed fast Fourier transform (FFT),[5] wherever possible. CosmoTools is triggered at selected time steps via a separate input deck in which the analysis tools and their parameters (e.g., the friends-of-friends (FOF) halo finder linking length) are specified. Given the performance requirements, we stress that it is not possible to straightforwardly integrate a number of commonly used community tools and scale them up to the run sizes needed for a simulation like the Last Journey. Therefore, the continued development of CosmoTools has been a crucial step in enabling the work described in this paper. As an example of the analysis output from CosmoTools, Figure 1 shows the result of a zoom-in to the largest halo in the Last Journey with a mass of $m_h \sim 6 \cdot 10^{15} h^{-1}\ M_\odot$ measured with an FOF halo finder using a linking length of $b = 0.168$, shown in the lower right corner of the figure.

A major goal for the Last Journey simulation is to provide a base data set for generating detailed synthetic sky catalogs. The generation of such catalogs is a highly nontrivial task, and no single method has proven to be decisively superior over others so far. For the Outer Rim simulation, we employed a hybrid approach by combining empirical and semianalytic modeling (SAM), with a much larger weight given to the empirical modeling, resulting in the cosmoDC2 extragalactic catalog (Korytov et al. 2019). One reason we took this route was that Galacticus (Benson 2012), the SAM we chose to use, is computationally expensive and therefore difficult to carefully tune to match observational constraints. For the Last Journey simulation, we plan to use L-GALAXIES[6] (see Henriques et al. 2015 and references therein), another publicly available SAM with somewhat less reliance on empirical models. This work will be described in a future paper. In addition to generating galaxy catalogs, we will also compute weak lensing information to provide a complete set of synthetic maps for dark energy studies with optical surveys and a range of data products geared toward ongoing and upcoming cosmic microwave background (CMB) experiments, which will also facilitate cross-correlation studies. Throughout the paper, we will highlight the specific data products generated to enable these goals.

The paper is organized as follows. In Section 2, we describe the simulation specifications, including the cosmology, simulation volume, and mass resolution. First-level outputs from the simulation are then described in Section 3. Next, we list the data products that are obtained in postprocessing in Section 4. We show selected results in Section 5, including measurements of power spectra, mass functions, and the halo concentration–mass relation. The results are in good general agreement with previous work, confirming the quality of the simulation. As part of this paper, we release some of the data products and describe those in Section 6. Finally, we present a conclusion and provide an outlook in Section 7.

2. Simulation Specification

The Last Journey simulation evolved $10{,}752^3$ particles (∼1.24 trillion) in a $(3400\ h^{-1}\,\mathrm{Mpc})^3 = (5025\,\mathrm{Mpc})^3$ volume. We have chosen a best-fit Planck cosmology (Planck Collaboration et al. 2020, Table 2, base-ΛCDM fit combining CMB spectra, with CMB lensing reconstruction and baryon acoustic oscillation (BAO) measurements), a ΛCDM model with massless neutrinos,

$$\Omega_{\mathrm{cdm}} = 0.26067, \tag{1}$$

$$\omega_b = 0.02242 \Rightarrow \Omega_b = 0.049, \tag{2}$$

$$h = 0.6766, \tag{3}$$

$$\sigma_8 = 0.8102, \tag{4}$$

$$n_s = 0.9665, \tag{5}$$

$$w = -1, \tag{6}$$

resulting in a total matter contribution of $\Omega_m = 0.310$. This leads to a mass resolution of $m_p \sim 2.7 \cdot 10^9 h^{-1}\ M_\odot$. The simulation was started at $z_{\mathrm{in}} = 200$ using the Zel'dovich approximation (Zel'dovich 1970) for setting up the initial conditions. We used the publicly available code CAMB (Lewis et al. 2000) to generate the input transfer function. A detailed description of the initial condition generation is given in Heitmann et al. (2010). In the same paper, the use of the Zel'dovich approximation versus higher-order schemes is discussed in detail and carefully tested. It was found that if

[5] A stand-alone version of HACC's distributed-memory, pencil-decomposed, parallel 3D FFT, SWFFT, is available at the following URL: https://xgitlab.cels.anl.gov/hacc/SWFFT.

[6] http://galformod.mpa-garching.mpg.de/public/LGalaxies/

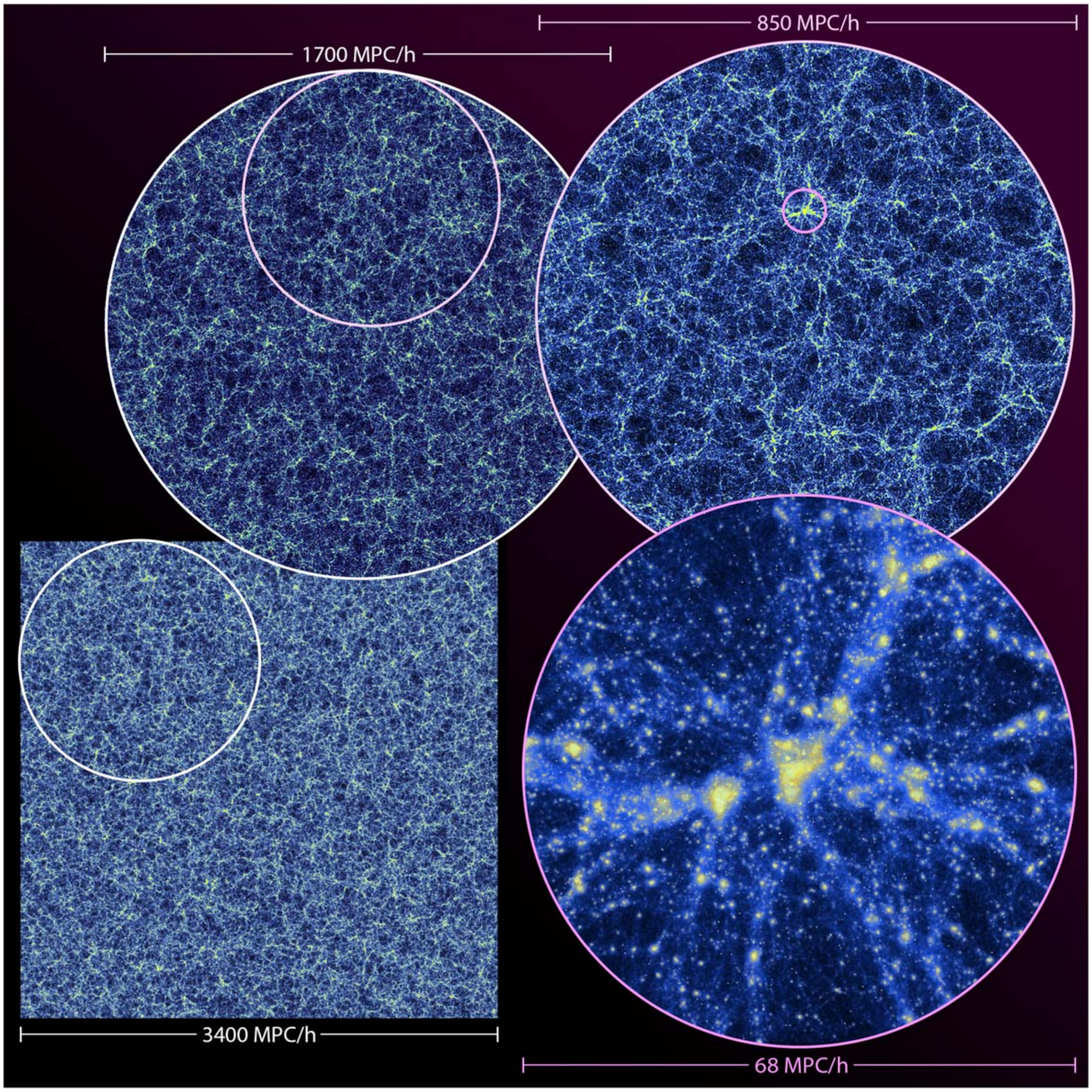

Figure 1. Visualization of the Last Journey simulation. Shown are thin density slices for the full box (lower left corner) and zoom-ins at different levels. The panel in the lower right corner focuses on the largest cluster in the simulation with a mass of $\sim 6 \cdot 10^{15} h^{-1} M_\odot$.

the simulation is started at a high enough redshift and in a large volume such that the initial move of the particles is a fraction of the initial particle interspacing, the Zel'dovich approximation provides highly accurate initial conditions, basically indistinguishable from higher-order schemes. In addition, a high-redshift start ensures that structures have enough time to form correctly at the scales of interest. We followed the findings in Heitmann et al. (2010) to set up very accurate initial conditions for the Last Journey simulation. The simulation was carried out with HACC running in its tree-PM mode, with a force resolution of $3.16 h^{-1}$ kpc. The high mass and force resolution in combination with the large volume will enable the creation of detailed synthetic sky catalogs across wave bands, including weak lensing maps, cluster weak lensing maps, optical cluster modeling, and galaxy clustering, to name just a few. A wide range of different modeling approaches can be employed for the generation of such catalogs due to the excellent resolution provided by the Last Journey simulation.

3. Outputs

Outputs from the HACC simulation consist of raw particle information (full and subsampled), as well as derived quantities. In order to reduce the amount of data generated, we employ two strategies: first, we compress the data in a lossless fashion using the Blosc library,[7] and second, we carry out as much of the analysis as possible in situ to avoid storing raw data files. The compression of particle files that contain mostly positions and velocities leads to a reduction of data by a factor of 1.4. For outputs that are dominated by integers (e.g., files that contain halo and particle tags), the compression factor can be as high as 7×. The compression rates are at the expected level, as detailed in the following. The particle data comprise 36 bytes particle^{-1}: 12 bytes for the velocities (which do not compress very well), 12 bytes for positions (for which the sign and exponent will compress well and the lower mantissa bits will not, leading to a target of 72% compressed size), and the

[7] https://blosc.org/

remainder essentially metadata (particle IDs, mass, mask) that almost completely compress. Thus, ideally, we would obtain an overall 57% compression rate for the particle files via $(12 + 12 \times 0.72)/36$. The mostly regular information does not compress perfectly (because there is some information content), and the compressor is not perfect, so 1.4× is in the expected range. The total amount of analysis data generated was ~720 TB. We also stored five full snapshots (~150 TB) of the full raw particle data for possible postanalysis studies.

As in previous large-scale simulations carried out with HACC, we store outputs at 101 time snapshots between $z = 10$ and 0, evenly spaced in $\log_{10}(a)$ for most of our data products. This leads to the following output values in redshift:

$$\begin{aligned} z = \{&10.04,\ 9.81,\ 9.56,\ 9.36,\ 9.15,\ 8.76,\ 8.57,\ 8.39,\ 8.05,\\ &7.89,\ 7.74,\ 7.45,\ 7.31,\ 7.04,\ 6.91,\ 6.67,\ 6.56,\ 6.34,\ 6.13,\\ &6.03,\ 5.84,\ 5.66,\ 5.48,\ 5.32,\ 5.24,\ 5.09,\ 4.95,\ 4.74,\ 4.61,\\ &4.49,\ 4.37,\ 4.26,\ 4.10,\ 4.00,\ 3.86,\ 3.76,\ 3.63.3.55,\ 3.43,\\ &3.31,\ 3.21,\ 3.10,\ 3.04,\ 2.94,\ 2.85,\ 2.74,\ 2.65,\ 2.58,\ 2.48,\\ &2.41,\ 2.32,\ 2.25,\ 2.17,\ 2.09,\ 2.02,\ 1.95,\ 1.88,\ 1.80,\ 1.74,\\ &1.68,\ 1.61,\ 1.54,\ 1.49,\ 1.43,\ 1.38,\ 1.32,\ 1.26,\ 1.21,\ 1.15,\\ &1.11,\ 1.06,\ 1.01,\ 0.96,\ 0.91,\ 0.86,\ 0.82,\ 0.78,\ 0.74,\ 0.69,\\ &0.66,\ 0.62,\ 0.58,\ 0.54,\ 0.50,\ 0.47,\ 0.43,\ 0.40,\ 0.36,\ 0.33,\\ &0.30,\ 0.27,\ 0.24,\ 0.21,\ 0.18,\ 0.15,\ 0.13,\ 0.10,\ 0.07,\ 0.05,\\ &0.02,\ 0.00\}. \end{aligned} \tag{7}$$

3.1. Snapshots

We save downsampled particle outputs at the 101 redshifts specified above. These snapshots can be used to generate light cones with different observer positions, estimates of correlation functions at specific redshifts, and coarse-grained density maps at specific redshifts. We randomly save 1% of the total number of particles in each snapshot where the choice of selected particles is kept the same after the initial snapshot. Overall, the storage requirement for these files is ~30 TB. In addition to the downsampled particles, we also saved a small number of full outputs at the following redshifts: $z = 2.02$, 1.01, 0.58, 0.15, and 0.0. The storage requirement for these files is ~150 TB.

3.2. Particle Light Cones

Below redshift $z = 10$, particle light cones are generated and output during each simulation time step. We store a full sphere relative to an observer placed in the corner of the simulation box at (0, 0, 0) for the generation of full-sky maps and realizations of smaller area outputs as required. For the redshift range $3 \leqslant z \leqslant 10$, the outputs are saved as light-cone particles downsampled to 1% of the full particle count. For redshifts $z < 3$, we save full particle light cones. In addition to positions and velocities, we store the potential from the particle mesh (PM) solver interpolated to the particle position on the light cone. This information will enable the reconstruction of a (coarse-grained) tidal field to, e.g., carry out future intrinsic alignment (IA) studies. The full light-cone outputs out to $z \sim 3$ result in ~286 TB of data, whereas the downsampled particle outputs to $z \sim 10$ result in 7 TB of data.

At high redshifts, the volume of the simulation is not sufficient to cover the full sphere. At redshift $z = 10$, we therefore replicate the simulation volume 64 times (without rotations) to cover a comoving distance of $6519.52h^{-1}$ Mpc, equivalent to the radius of the light-cone sphere. The large memory overhead of the replication requires downsampling of the particles to enable the in situ evaluation of the light cone keeping within the computer's memory. The downsampling also reduces the storage footprint required for the light cone. For the science cases of interest, e.g., CMB lensing, the downsampled light cones at high redshift are sufficient to generate maps at the required accuracy. At lower redshifts, the number of replicants reduces accordingly with the shrinking light-cone sphere, allowing us to save the full particle light cone for $z < 3$.

This method of in situ particle light-cone generation has several benefits over postprocessing from particle outputs. It removes the need for particle outputs at every analysis step, which are expensive in terms of both storage and I/O. Additionally, it is well known that extrapolation or interpolation between analysis steps causes small-scale inaccuracies in light cones when orbital timescales are comparable to or smaller than the interstep spacing (see, e.g., Kitzbichler & White 2007; Merson et al. 2013; Korytov et al. 2019). While neglecting the evolution between analysis steps can reduce these small-scale errors, it introduces discretization inaccuracies into the clustering. The in situ computation uses extrapolation of the position and velocity data within the simulation time-step integrator, effectively setting the time resolution of the light-cone computation to be similar to that of the simulation itself. In this way, the light-cone inaccuracies and postprocessing requirements are significantly reduced compared to the postprocessing computation.

3.3. Halos

Next, we describe the halo files that are generated as part of the Last Journey simulation. They provide the foundation for the creation of synthetic sky maps. Halo information enables detailed studies of structure formation processes and investigations of the galaxy–halo connection. We have run an FOF halo finder and determined halo spherical overdensity (SO) properties at the 101 snapshots listed in Equation (7). The FOF finder is based on the standard definition first introduced for cosmological applications in Davis et al. (1985) and follows a fast tree-based algorithm (Hsu et al. 2010). We use the centers of the FOF halos (given by their local potential minima) to grow SO halos. This approach allows for efficient identification of SO halos and generally avoids overlaps between halos. However, one concern might be that due to the irregular shapes of the FOF halos and the linking of halos that can occur in the FOF approach, small halos might be missed. Knebe et al. (2011) carried out an extensive halo finder comparison project, including the halo finder that is used here (the LANL halo finder in Knebe et al. 2011). They found excellent agreement between the LANL and other SO finders if the SO halos are well sampled with at least 500 particles,[8] the same mass cut we employ here for the SO halos. The in situ halo information results in close to 200 TB of data. We have stored the following files and information.

1. Halo properties. We measure halo properties for both FOF and SO halos. The mass cut on FOF halos is 20 particles halo^{-1}, leading to a minimum halo mass of

[8] In Knebe et al. (2011), a linking length of $b = 0.2$ was used. The overlinking problem is less severe with a smaller linking length. In this paper, we use $b = 0.168$. Together with our minimum mass of 500 particles for the SO halos, this choice ensures that we capture the SO halos reliably.

$m_{\rm FOF} \sim 5.4 \cdot 10^{10} h^{-1}\ M_\odot$. We use a linking length of $b = 0.168$. This value is commonly used to create halo occupation distribution (HOD)–based mock catalogs for, e.g., redshift space distortion studies (Reid & White 2011), measurements of the clustering of massive galaxies in the BOSS survey (White et al. 2011), and several recent eBOSS studies (Zarrouk et al. 2018; Gil-Marín et al. 2018; Hou et al. 2018; Avila et al. 2020; Alam et al. 2020; Smith et al. 2020; Rossi et al. 2020). We find SO halos for each FOF halo that has at least 500 particles. In addition, for five snapshots between $z = 0$ and 2.02, we measure FOF halo properties for a linking length of $b = 0.2$ to enable mass function comparisons with results available in the literature (see Section 5.4 for details).

1. For the FOF halos, we save the number of particles in a halo, the halo tag, the halo mass in units of $h^{-1}\ M_\odot$, the kinetic energy, the halo centers as measured by the local potential minimum and the center of mass, the halo angular momentum, the halo circular velocity, the halo velocity, the halo velocity dispersion, and the eigenvectors of the halo inertia tensor.
2. For the SO halos, we save $M_{200\,c}$ measured as the number of particles in units of $h^{-1}\ M_\odot$, the halo radius (r_{200c} in units of h^{-1} Mpc), the kinetic energy, the velocity dispersion, the circular velocity, the angular momentum, the halo velocity, the inertia tensor eigenvectors, and the concentration measured in three different ways (profile fitting, accumulated mass, and peak measurement; see Child et al. 2018 for details).

2. Halo particle tags. The halo particle tag files contain tags for all particles that reside in halos and their halo tag. The main use case for this output is the construction of merger trees in postprocessing.
3. Halo particles. We store all particles in halos that have at least 10,000 particles, translating into a halo mass of $m_h = 2.7 \cdot 10^{13} h^{-1}\ M_\odot$. We also store randomly selected particles from all halos downsampled at a rate of 1%, with at least five particles per halo.
4. SO mass profiles. For all SO halos, we store their mass profiles binned in 20 radial bins.

3.4. Halo Cores

In addition to identifying halos and determining their properties at each analysis time step, we also save halo "core" information over time. Halo cores, as described further below, enable tracking of halo substructure information. For each FOF halo with at least 80 particles, we store the halo core, defined here as the 50 particles closest to the potential center of the halo. These particles are saved in core particle files for each of the 101 analysis time steps. The raw files add up to ∼32 TB of storage. Additionally, once a particle has been identified as a core particle, it is tracked and recorded until the last analysis time step, independently of whether it is in a halo or not in the next analysis step. These accumulated core files add up to almost another 150 TB. The core and accumulated core files allow us to track halo substructure by following halo core particles that have fallen into more massive host halos, serving as an alternative to subhalo finding and tracking. In a postprocessing step, we generate a core catalog as described in Section 4.2. The core catalog contains information about the cores themselves, such as sizes, positions, and velocities, as well as their temporal history and later role inside a halo (central versus satellite) and the halo properties at infall (when merging into a larger halo). Once the core catalog is generated, the core and accumulated core files do not need to be stored.

4. Postprocessing Analysis

From the information saved during the in situ analysis steps described in Section 3, several additional data products are generated in postprocessing. These data products are used to build synthetic sky catalogs in different wave bands and track the evolution of structure over time. The approach we are taking is customized for the HACC analysis infrastructure to ensure the required scalability and speed needed when analyzing very large simulation outputs. In the following, we provide a brief description of each of the data products and their potential usage in follow-up science investigations.

4.1. Merger Trees

Merger trees describe the hierarchical formation of halos and subsequent mass accretion by computing the overlap in halo particles from adjacent catalog snapshot files. They provide the means to gain a more fundamental understanding of the hierarchical nature of structure formation, as well as a framework within which to model galaxy formation in the cosmological context. In particular, merger trees are essential for investigating the galaxy–halo connection if one desires to go beyond simpler approaches based only on individual halo properties (for a recent review of the galaxy–halo connection, see Wechsler & Tinker 2018 and references therein). The development of synthetic sky catalogs using methods that rely on detailed information on halo and substructure evolution, such as SAMs, is also based on merger trees. Merger tree construction is a nontrivial process, and different methods have been developed in the literature (for a comparison study and description of popular merger tree codes, see Srisawat et al. 2013).

Challenges in merger tree building include mass-resolution effects and halos that "split," i.e., halos that may have multiple descendants in subsequent snapshots. Finite mass resolution (and the connected minimum halo mass) can have a stochastic thresholding effect leading to halos that disappear from one snapshot to the next. This problem has long been recognized, and different mitigation schemes have been discussed in the literature for both halo and subhalo merger trees (see, e.g., Fakhouri & Ma 2008; Behroozi et al. 2013; Rodriguez-Gomez et al. 2015; Han et al. 2018 and references therein). Instead of inserting "phantom" halos to account for disappearances, we mitigate this problem by using a soft threshold. Once a halo has been found because the particle count crossed the desired threshold (e.g., 50 particles), we can allow that halo to continue to exist in subsequent analysis steps even if its mass dips below the original preset value to some new minimum value (e.g., 20 particles).

Halo splitting commonly occurs in simulations, as neighboring FOF halos can occasionally be overlinked and identified as a single object, even when they have not dynamically merged. In later time steps, these halos may be rediscovered as separate objects. Moreover, some of these halos are involved in "flybys" and never merge after splitting. These complications can make halo temporal connectivity ill-defined. To address splitting, we

Table 1
Properties in the Core Catalog File

Property	Description
x, y, z	Position of central core particle
v_x, v_y, v_z	Velocity of central core particle
Radius	Core radius
Velocity dispersion	Velocity dispersion of core particles
Core tag	Unique label of core (assigned at core creation and kept unmodified throughout catalog)
Tree node index	Unique identifier of merger tree object containing core at current time step
Central/satellite	1 if a central core and 0 if a satellite core; central cores are assigned properties of the host merger tree object
Host core tag	Core tag associated with central object at time of infall (same as core tag for centrals)
Infall step	Simulation time step immediately preceding infall (current step if central)
Infall halo properties	All host halo quantities recorded at infall (for centrals, it is the current step halo properties)

utilize the halo "fragmentation" approach described below, intended to track all identified mass objects in the simulation by artificially breaking up halos that are later identified to have split.

Our merger tree implementation is focused on good scalability, efficient memory usage, and high accuracy. A detailed description of our approach, with an emphasis on algorithmic details, is given in Rangel et al. (2016); we provide a brief summary here. We stress that we only describe the construction of halo merger trees in this section and not subhalo merger trees, commonly used for the analysis of high-resolution simulations. To track substructures (structures that have fallen into a halo), we utilize our newly developed core-tracking approach, introduced in Rangel et al. (2016) and applied to the modeling of galaxy distributions in D. Korytov et al. (2021, in preparation). We summarize the core methodology in Section 4.2.

To facilitate building merger trees, we store the tags of all particles that reside in halos and the corresponding halo tag at each of the 101 analysis steps. In addition, we store all halo properties in the halo catalog, queryable using the same halo tags. This information allows us to track the evolution of all halos via their particle content. The construction of the merger trees starts with the last snapshot at $z = 0$ and then works backward in time until the first analysis snapshot is reached at $z \approx 10$. We identify overlapping halo particles between adjacent snapshots to determine the progenitors of each descendant halo. If a halo has an overlapping particle set above the minimum halo mass with multiple descendant halos (indicating the halo had split), we break up the halo into corresponding fragments. The original halo mass is assigned to the fragments proportionally by taking the ratio of each fragment intersection and their union (conserving total mass). As a result, the descendant of every halo is the halo with the greatest amount of particle overlap.

The fragment halos serve as markers for mass objects with individual properties not captured in the catalog, as their existence was hidden due to numerical effects such as overlinking. We note the advantage of temporally constructing the tree in reverse time. All of the objects at $z = 0$ serve as the base halos (final descendants) of every tree. Walking backward, we can connect halos and construct artificial fragments as needed in a connectivity tree that consistently tracks all mass that eventually ends up in the final objects of interest at $z = 0$. Flyby halos, by construction, will not appear in the final tree, as they never ended up in the final mass at $z = 0$. Constructing trees forward in time would require continuously breaking up trees during every split encountered and can significantly complicate merger tree construction (often requiring multiple passes of tree pruning to correct). In summary, our approach produces clean merger trees, where all halos at any given redshift have one descendant (no splits), and for each halo, we have amassed its entire connectivity history.

4.2. Core Catalog Generation

The core catalog is generated from three input data sets: the core particle files, accumulated core files, and merger trees. Incorporating the merger tree information allows us to track not only halos but also cores over time in the core catalog. We also include information from the halo property files for each core. Table 1 summarizes the information stored in the core catalog. In many ways, the core catalog is the main analysis output from the simulation, in that it encapsulates the history and properties of each halo in the simulation. In addition to the merger tree data, it provides information about the halos after they have fallen into other halos and therefore can be substituted as a proxy for a subhalo catalog.

Subhalo catalogs are used for a wide range of investigations, including the study of substructure in cluster-sized halos in gravity-only simulations (e.g., Gao et al. 2012) and hydrodynamic simulations (e.g., Nagai & Kravtsov 2005), or as proxies for galaxy positions (see, e.g., the investigations in Wetzel et al. 2009). In combination with merger tree information, they are the base ingredient for SAMs and subhalo abundance modeling approaches. Onions et al. (2012) carried out an extensive subhalo finder comparison project and found good agreement between different approaches. The subhalo finding approach comes with several associated challenges. First, for a simulation of the size of the Last Journey run, algorithms that go beyond simple spatial analysis are expensive. In particular, given that the subhalo finder has to be run on the order of 100 snapshots, the computational burden is considerable. Second, the demands on mass resolution are very severe for enabling accurate subhalo tracking. Third, the distribution of subhalos close to the center of the parent halo does not allow for correct placement of galaxies. This is due to the disruption of subhalos when they are close to the halo center. In order to overcome these challenges, we generate core catalogs that enable the tracking of the evolution of particles that at some point belonged to the inner core of a halo. This approach is a natural extension of the ideas in Kaiser (1984) and White et al. (1987) and is related to Hong et al. (2016), where the most-bound particle in a halo was tracked and used to build merger trees. The point of using multiparticle cores (as enabled by high mass resolution simulations) is to provide a potentially robust way to study the evolution of halo substructure over time.

A detailed description of the core tree assembly is given in D. Korytov et al. 2021 (in preparation). (The core catalog contains all of the core trees.) Here we provide a brief summary. As explained in Section 3.4, we identify in each analysis step the 50 particles that are closest to halo centers for halos with more than 80 particles and continue to track their evolution over time. For each core extracted from these files, we measure its properties, including position and velocity (taken from the centermost particle of the core), radius (defined as the rms of the positions), and velocity dispersion.

Cores are marked with multiple tracking identifiers. First, we report a "core tag," a unique ID that does not change over time once it is assigned (i.e., when a core was first identified). The core tag is based on the merger tree node index of the halo associated with the newly birthed core.

Next, we note whether a core is a central core or a satellite; the latter terminology indicates that the core host halo merged with another larger halo, after which the core of interest is considered to be a satellite in the new halo. While constructing the core catalog, every time halos are recorded as merged in the merger tree, the absorbed halos will have their central cores converted into satellites, with the descendant halo inheriting its central core from the most massive progenitor. Moreover, the satellites that existed prior to the merger in all of the progenitors will be carried into the merged descendant as well; thus, a given halo at any snapshot only has one central core, with the rest of its substructure encapsulated by all of the satellites it has accrued over all of its previous merger history.

Note that, unlike the merger tree, which was constructed backward in time, the core trees are created forward in time and track core history via the discussed merging criteria. While building the core trees, the central core properties are extracted from the core particle files, as they are measured for every halo on every analysis snapshot. Satellite halo properties are determined separately. As all satellites were originally centrals of a halo that merged, during catalog construction, we temporarily store the central core particles of all merged halos at infall. Using this stored particle set, satellite properties at any following snapshot postinfall are then extracted from the accumulated core particle files, which we recall have been tracking up-to-date properties of any particle ever identified as being in a core.

The second core tag we record in the catalog is referred to as the "tree node index," which indicates the hosting halo of the core in the given snapshot. If the core is a central, the core tag and tree node index are the same. If the core has merged into another halo and, therefore, is a satellite, the tree node index will indicate the new parent halo. The final identifier is the "host core tag." This tag is important when a core falls into another halo; the host core tag then reports the core tag associated with the central object of its new host halo at the time of infall (a useful quantity to determine the hereditary order of a core as described in Section 5.6). For a core that remains a central throughout its history, either because it lives an undisturbed existence or because it is the core of a dominant halo that absorbs other halos but is never absorbed itself, all the tags mentioned so far will be identical.

We note that we include all halo properties in the core catalog directly to avoid having to perform matching procedures across files later. The overhead in storage is negligible. It is very important to carefully record halo properties at the time of infall for satellites (for centrals, these are simply their host halo properties). These properties can then be used to model the further evolution of the core inside its new host halo. For example, a simple mass-loss model can be applied to enable the use of cores in place of subhalos. We store all of the halo properties that are available and described in Section 3.3.

To summarize, all objects in the core catalog are grouped together with the tree node index (referencing merger tree objects) at the current step and can be traced in time using the core tag. The properties of the merger tree object itself are assigned to the central core, which in turn is marked by the central flag and has an infall step set to the current step. With some additional modeling, the information stored in the core catalogs enables the creation of substructure merger trees and, as a result, can be used as input for SAMs. We will discuss this idea further in a forthcoming paper.

4.3. Halo Light Cones

Dark matter–dominated halos form the base for synthetic galaxy catalogs, in which galaxies are typically placed according to either a profile or measured substructure. Halo light cones allow for the generation of galaxy catalogs in the frame of an observer. Alongside lensing information from particle light cones, these can be used to create synthetic observations of galaxy surveys such as that of Korytov et al. (2019). These require halo catalogs measured at regular steps throughout the simulation, as well as merger tree information on the temporal evolution of the halos.

We generate halo light cones in postprocessing from the halo catalog files and merger trees. We select merger tree halo objects from a given snapshot and interpolate their positions backward onto the light cone using their progenitor position at the previous snapshot. For objects with multiple progenitors, we choose the most massive progenitor of the halo for the progenitor position, and when no progenitors are found, we instead extrapolate the position backward onto the light cone using the velocity at the later snapshot. The case of fragment halos is treated separately; for each set of halo fragments, we retain the most massive fragment and discard all other objects associated with the FOF halo, again using the merger tree information to interpolate or extrapolate as above. Lastly, we match against the halo catalog at the later snapshot to obtain the properties for the FOF halo associated with this fragment object, including the corresponding SO quantities as well.

Following the procedure for the particle light cones, the halo light cones are computed for a full sphere relative to an observer placed in the corner of the simulation box at (0, 0, 0), thus enabling full-sky maps or several realizations of smaller area outputs. These are saved for the redshift range $z < 10$ with all SO and FOF halo properties as listed above. These halo light cones are used to generate survey-like catalogs from the simulations, as well as to access dark matter halo information in light-cone coordinates.

5. Selected Results

In the following, we show selected results obtained from the Last Journey simulation. We emphasize that more results will be published in forthcoming, focused papers. In order to demonstrate the high quality of the simulation, we display a set of standard measurements compared to fitting functions and emulators available in the literature. We provide results based

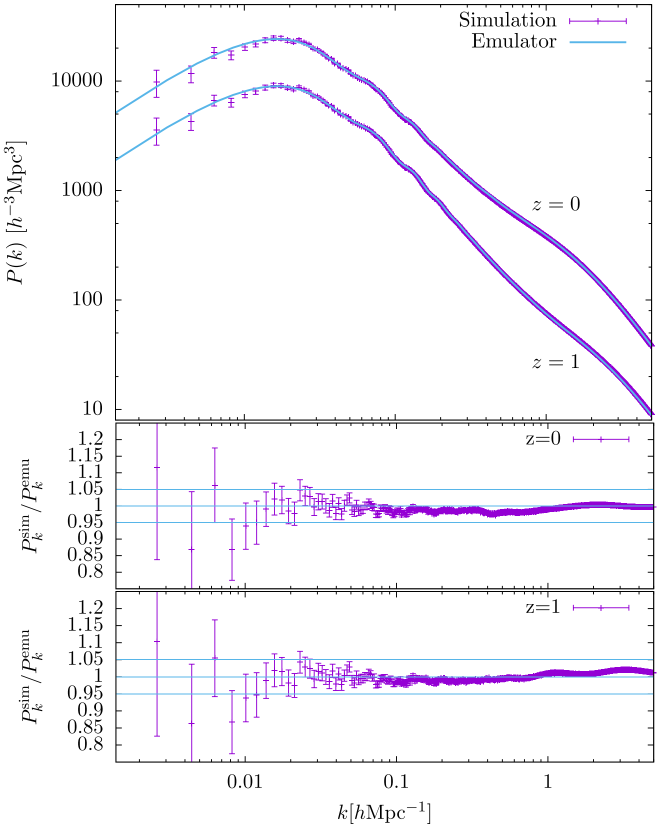

Figure 2. Upper panel: power spectrum results at redshift $z = 0$ and 1. For comparison, we show predictions from the Cosmic Emulator by Heitmann et al. (2014). Lower panels: ratio of the simulation and emulator at $z = 0$ and 1. The light blue bands show a 5% range. The results are within the accuracy bounds reported in Heitmann et al. (2014).

on particle snapshot and light-cone outputs, measurements from the halo catalogs, and, finally, the core catalog.

5.1. Matter Power Spectra

The nonlinear matter power spectrum is an important statistic for conveying cosmological information. From a simulation perspective, the matter power spectrum also provides a very good estimator for the accuracy of the simulation down to a certain length scale. Extensive work has been carried out in the past to compare power spectra from different codes (Heitmann et al. 2005, 2008; Schneider et al. 2016) and assess the accuracy at which the matter power spectrum can be obtained given certain simulation settings, such as volume, particle loading, and force resolution (Heitmann et al. 2010). The HACC automatically generates a power spectrum measurement every time it stores a checkpoint/restart file. Overall, we saved 55 power spectra from the full simulation. Figure 2 shows the measurement of the power spectrum from the Last Journey simulation at two redshifts, $z = 0$ and 1. In addition, we show predictions at those redshifts from the Cosmic Emulator (Heitmann et al. 2014). The agreement is excellent, well below 5%, the reported accuracy for the emulator. The comparison with the emulator is also an indirect comparison of GADGET-2 (Springel 2005) and HACC, given that the emulator used in this comparison is based on a large suite of GADGET-2 simulations. The error bars shown are those that would be expected for a Gaussian random field with the measured power spectrum, with the standard deviation of power in the ith wavenumber bin, $\sigma_{P(k_i)}$, given by

$$\sigma_{P(k_i)} = \frac{P(k_i)}{\sqrt{N_{\mathrm{modes}}(k_i)}}, \tag{8}$$

where $P(k_i)$ is the estimated power in the bin and $N_{\mathrm{modes}}(k_i)$ is the number of independent modes in the bin. These error bar estimates were checked against the variance in ensembles of discrete Gaussian random field realizations.

5.2. Matter Correlation Function

We also computed the two-point correlation function, the real-space analog of the power spectrum. This was performed by counting pairs on a 1% downsampled population of particles (approximately 12 billion), counting the total number of particle pairs DD_i that fall within the ith bin in separation distance, with the midpoint of the bin denoted by r_i. The correlation function is then computed using the Minus estimator,

$$\xi(r_i) = \frac{DD_i}{RR_i} - 1, \tag{9}$$

where RR_i is the equivalent number of particle pairs that would fall within the distance bin given a random distribution (see Equation (3.12) in Martínez & Saar 2002 or Equation (5) in Kerscher et al. 2000). As in our case, the geometry is a periodic box, and we compute this term analytically using

$$RR_i = NV_i\frac{N}{L^3}, \tag{10}$$

where L is the simulation box length, N is the number of particles used to count pairs for the DD values, and the volume of the spherical shell corresponding to the bin (of width Δ_i) is $V_i = \frac{4}{3}\pi\left[\left(r_i + \frac{1}{2}\Delta_i\right)^3 - \left(r_i - \frac{1}{2}\Delta_i\right)^3\right]$. An important result demonstrated by this calculation is the clear BAO signal at $\sim 100\ h^{-1}$ Mpc, as shown in Figure 3 at redshift $z = 0$, which plots the product of the measured correlation function and the square of the bin midpoints. The location and shape of the BAO bump is consistent with the overlaid model prediction using the power spectrum of Heitmann et al. (2014), converted to real space using the mcfit[9] Python package of Li (2019).

The error bars presented here are similar in scope to the power spectrum, but the error bars in correlation function space are highly correlated, and estimation is somewhat more involved. Following Sánchez et al. (2008), we start with the variance expected from the Gaussian component of density fluctuations and a shot-noise component from a finite tracer density (Feldman et al. 1994),

$$\sigma_P^2(k) = \frac{2}{V}\left(P(k) + \frac{1}{\bar{n}}\right)^2, \tag{11}$$

where V is the simulation volume and $\bar{n}$ is the mean tracer density. The covariance of the two-point correlation function is then given by Cohn (2006) and Smith et al. (2008),

$$C_\xi(r, r') = \int\frac{k^2\mathrm{d}k}{2\pi^2}j_0(kr)j_0(kr')\sigma_P^2(k), \tag{12}$$

[9] https://github.com/eelregit/mcfit/

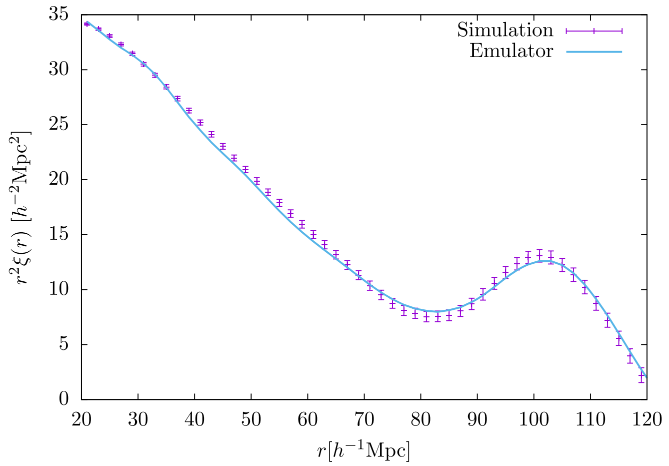

Figure 3. Real-space correlation function of downsampled particles at redshift $z = 0$ as computed from Equations (9) and (10). For comparison, we show a prediction based on the Cosmic Emulator by Heitmann et al. (2014). We leave this as a qualitative comparison, as the prediction was not derived directly from measuring correlation functions but via a Fourier transform of the power spectrum emulator, which makes the accuracy of the prediction difficult to quantify.

where j_0 is the zeroth-order spherical Bessel function, and the standard deviation of the two-point correlation function is then $\sigma_\xi(r) = \sqrt{C_\xi(r, r)}$. We used the 2D-FFTLog[10] Python package from Fang et al. (2020) to compute the integrals in Equation (12), and we verified the error bar estimates against a variance of the two-point correlation functions measured from several ensembles of discrete Gaussian random fields. A finite bin width will dilute the covariance, but for the binning presented here (bin width of $2\,h^{-1}$ Mpc from 20 to 130 h^{-1} Mpc), we found that the finite bin width reduces our error bar estimates by less than 3%, which is expected to be subdominant to other omitted effects, so we use Equation (12) directly in the thin shell approximation. We also found that the shot-noise component in Equation (11) comprises less than 0.1% of the errors for our density of tracer particles, so we conclude that our downsampled particle density is sufficient to fully resolve the measurement in the simulation box.

5.3. Angular Correlation Function

In order to verify the accuracy of our particle light-cone generation, we show the result for the measured angular correlation functions of two light-cone shells at redshifts $z = 1$ and 0.5. These are created using a nearest grid point density map on a HEALPix[11] (Górski et al. 2005) grid of Nside 8192, with the angular correlation function measured using Polspice[12] (Challinor et al. 2011) with a 15° apodization window (Figure 4). These are compared to the angular correlation function predicted from the Cosmic Emulator power spectrum, taking into account both the finite shell width and the apodization kernel. The agreement over the range $20 < \ell < 10,000$ is comparable to that of the matter power spectrum, with any light-cone extrapolation biases subdominant to the emulator errors.

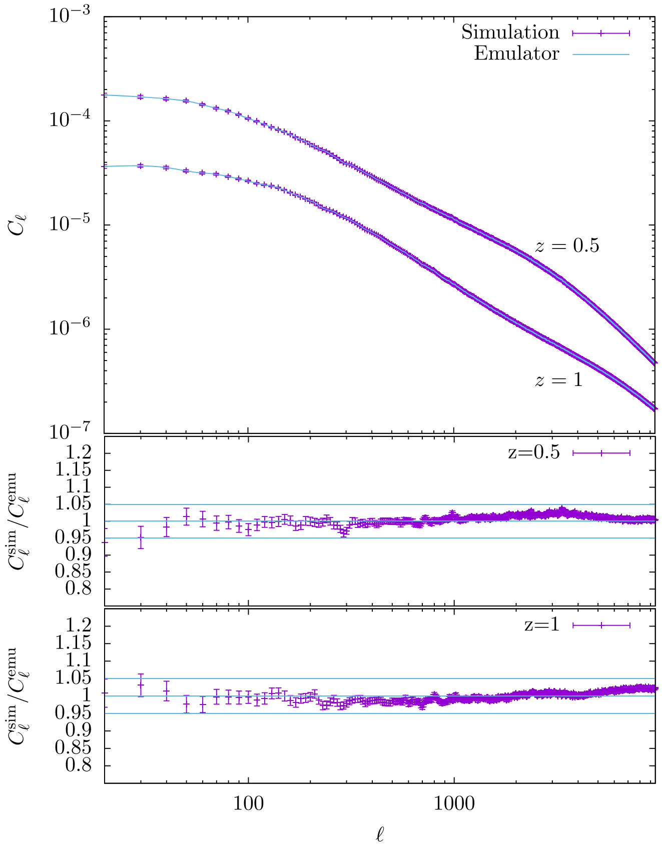

Figure 4. Upper panel: angular correlation function of particle overdensity in the light-cone outputs at $z = 0.5$ and 1.0 (see text for details). For comparison purposes, we show predictions computed from the Cosmic Emulator of Heitmann et al. (2014). We have chosen to plot every 10th multipole. The error bars show the 1σ predicted level of cosmic variance (neglecting non-Gaussian terms), corrected for the Polspice apodization kernel. Lower panels: ratio of the measurements and emulator predictions at $z = 0.5$ and 1.0, with light blue bands showing a 5% range. The results are consistent with the matter power spectrum shown in Figure 2 and within the accuracy bounds reported in Heitmann et al. (2014).

In angular space measurements, finite box lengths typically lead to errors at large scales and high redshifts, as displayed by, e.g., Figure 1 of Blaizot et al. (2005). This is attributed to the presence of correlations due to repeated structures along the line of sight, as well as in the perpendicular direction for large angular scales. Given the large box size of the simulation, we choose not to apply box rotations in order to avoid discontinuities; as a result, a pencil beam along the axis will be replicated for the first time at $z \approx 1.8$ and the second time at $z \approx 12$, while very large-scale measurements will begin to experience replications at $z \approx 0.7$. Correspondingly, correlations measured at high redshifts on large scales require further characterization and are ultimately limited by the finite box length of the simulation.

5.4. Halo Mass Function

Next, we show measurements of the halo mass function. We focus on the FOF mass function obtained by using a linking length of $b = 0.2$ and on predictions for overdensity masses M_{200c}. While we use a shorter linking length of $b = 0.168$ for the main analysis of the simulations, the measurements employing the larger linking length have been widely studied in the literature. An important result due to Jenkins et al. (2001) is that for this particular mass definition, the differential mass

[10] https://github.com/xfangcosmo/2DFFTLog/
[11] https://sourceforge.net/projects/healpix/
[12] http://www2.iap.fr/users/hivon/software/PolSpice/

function $f(\sigma, z)$,

$$f(\sigma, z) = \frac{d\rho/\rho_b}{d\ln\sigma^{-1}} = \frac{M}{\rho_b(z)}\frac{dn(M, z)}{d\ln[\sigma^{-1}(M, z)]}, \tag{13}$$

is close to universal. In this expression, $n(M, z)$ denotes the number density of halos with mass M, $\rho_b(z)$ is the background density at redshift z, and $\sigma(M, z)$ is the variance of the linear density field smoothed with a top-hat filter. Universality in this context means that a specific fitting function that encapsulates cosmology only through the linear power spectrum describes the mass function accurately across a range of redshifts. In this paper, we use the fitting function derived by Bhattacharya et al. (2011),

$$f^{\rm Bhatt}(\sigma, z) = A\sqrt{\frac{2}{\pi}}\exp\left[-\frac{a\delta_c^2}{2\sigma^2}\right]\left[1+\left(\frac{\sigma^2}{a\delta_c^2}\right)^p\right]\left(\frac{\delta_c\sqrt{a}}{\sigma}\right)^q, \tag{14}$$

with the parameters

$$A = \frac{0.333}{(1+z)^{0.11}},\quad a = \frac{0.788}{(1+z)^{0.01}},\quad p = 0.807,\quad q = 1.795. \tag{15}$$

The density threshold for spherical collapse, $\delta_c = 1.686$, is held fixed at all redshifts. In Heitmann et al. (2015), the fit included a small modification to simplify the redshift dependence: a was simply set to 0.788, eliminating the extra z-dependence. We compared our results to both fits and found that for $z = 1$, the simplified version performed slightly better, while for $z = 2$, the original form provided a closer fit.

In order to account for FOF discreteness effects for low-mass halos, we apply the empirical correction given by

$$n_h^{\rm corr} = n_h(1 - n_h^{-0.65}), \tag{16}$$

where n_h denotes the number of particles in a halo. A correction of this type was first introduced by Warren et al. (2006) and later recalibrated by Bhattacharya et al. (2011). We impose a halo mass cut and only include halos with masses larger than $10^{12}h^{-1}\ M_\odot$ ($\sim$400 particles halo^{-1}). At this point, the correction is very small. In order to guarantee good sampling per mass function bin, we require each bin to have at least 100 halos. This restricts our measurement range slightly in the cluster regime. The error bars shown are Poisson error bars. As discussed in Lukić et al. (2007), a slightly modified Poisson error given by $\sigma_\pm = \sqrt{n_h + 1/4} \pm 1/2$ introduced by Heinrich (2003) can be used if n_h in some bins is very small. However, since we require at least 100 halos bin^{-1}, this modification is negligible, and the familiar form $\sigma_\pm = \sqrt{n_h}$ can be used to accurately determine the error bars.

The measurements for the differential mass function $f(\sigma)$ are shown in Figure 5 for five redshifts between $z = 2.02$ and 0. The lower panel shows the ratio with respect to the fit given in Equation (14). The agreement is better than 5%–10% over most of the mass range, consistent with previous works. The discrepancy at higher masses is due the limited volume available from a single simulation. The fit derived in Bhattacharya et al. (2011) is based on more than twice the volume available in the Last Journey simulation and is also based on a different cosmology. For a comprehensive discussion about the volume effects on the mass function, see, e.g., Crocce et al. (2010).

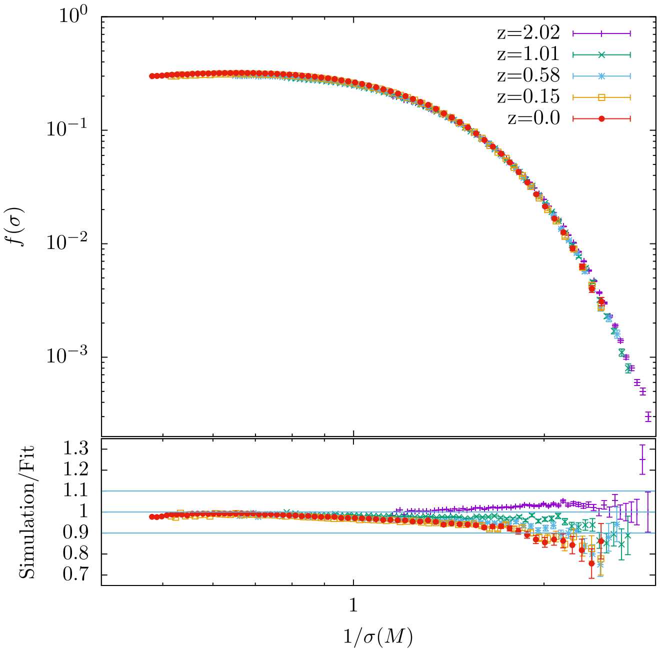

Figure 5. Upper panel: results for the differential mass function between redshifts $z = 0$ and 2 for FOF halos with a linking length of $b = 0.2$. We show $f(\sigma, z)$ as a function of $1/\sigma$ to show the level of universality found as a function of redshift. Lower panel: ratio of the simulation with respect to the mass function fit by Bhattacharya et al. (2011). The light blue lines show 10% boundaries. Over a wide range, the agreement is well below 10% discrepancy; at the higher mass, the agreement degrades slightly. This is in good agreement with previous studies of single large boxes from, e.g., the Outer Rim simulation (Heitmann et al. 2019b).

Turning to the SO mass definition, Figure 6 shows a comparison between the simulation results at $z = 0$ and 1 with respect to a recently developed mass function emulator (Bocquet et al. 2020) based on the Mira–Titan Universe simulation suite, first described in Heitmann et al. (2016). The Mira–Titan Universe includes a dynamical dark energy equation of state and neutrino mass, in addition to the standard six cosmological parameters. The emulator's mass range is limited by the lower mass resolution of the Mira–Titan Universe simulations (around $m_p \sim 10^{10}\ M_\odot$, depending on the cosmology for each model) and provides predictions at the group and cluster scales, $m_{\rm halo} \geqslant 10^{13}h^{-1}\ M_\odot$. Overall, the agreement between the simulation and the emulator for both redshifts is around 2%–3%. As for the differential mass function, we employ Poisson error bars in Figure 6.

5.5. Concentration–Mass Relation

We present results for the halo concentration–mass relation for two redshifts, $z = 0$ and 1. We measure the concentration by fitting a Navarro–Frenk–White (NFW) profile (Navarro et al. 1996, 1997) to each halo, as given by

$$\rho(r) = \frac{\delta\rho_{\rm crit}}{(r/r_s)(1 + r/r_s)^2}, \tag{17}$$

where δ is a characteristic dimensionless density, and r_s is the scale radius of the NFW profile. The concentration of a halo is defined as $c_\Delta = r_\Delta/r_s$, where Δ is the overdensity with respect to the critical density of the universe, $\rho_{\rm crit} = 3H^2/8\pi G$, and r_Δ

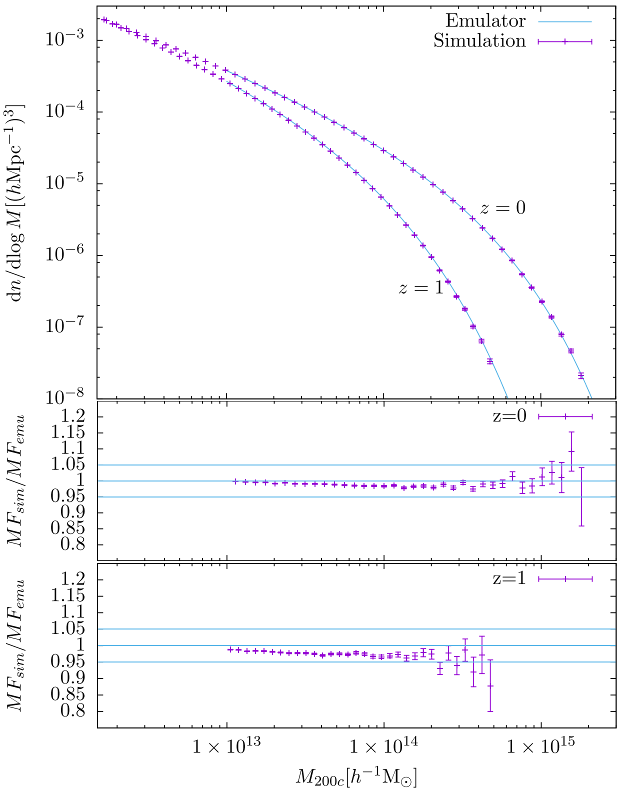

Figure 6. Upper panel: mass function results at redshift $z = 0$ and 1 for SO halos, M_{200c}. For comparison, we show predictions from the Cosmic Emulator (Bocquet et al. 2020). Lower panels: ratio of the simulation and emulator at $z = 0$ and 1. Note that the emulator predictions start only at masses larger than $10^{13}h^{-1}\ M_\odot$, therefore reducing the mass range shown in the ratio panels. The light blue bands show a 5% range. The results are within the accuracy bounds reported in Bocquet et al. (2020).

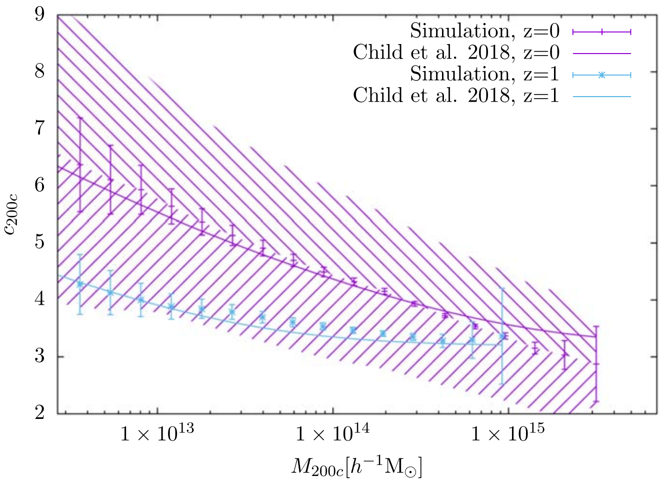

Figure 7. Halo concentration–mass relation measurement for redshift $z = 0$ (purple) and 1 (light blue). For $z = 0$, we show the 1σ standard deviation as the hatched region. In addition to the measurements from the simulation, we show the fitting function derived in Child et al. (2018; their Equation (18)), see also our Equation (18).

is the radius at which the enclosed mass, M_Δ, equals the volume of the sphere times the density, $\Delta\rho_{\mathrm{crit}}$. We compute concentrations corresponding to $\Delta = 200$, corresponding in turn to $c_{200c} = R_{200c}/r_s$. Details about our fitting approach are given in Bhattacharya et al. (2013) and Child et al. (2018). Child et al. (2018) also discussed additional approaches for measuring halo concentrations and compared them. Following the results found in that paper, we use the profile fitting technique here, which provides robust measurements. We compare our results to an approximate fit derived in Child et al. (2018; see their Equation (18) and Table 1),

$$c_{200c} = A\left[\left(\frac{M_{200c}/M_\star}{b}\right)^m\left(1 + \frac{M_{200c}/M_\star}{b}\right)^{-m} - 1\right] + c_0, \tag{18}$$

with $A = 3.44$, $m = -0.1$, and $c_0 = 3.19$ for $z = [0, 4]$ and measurements for all (relaxed and unrelaxed) halos. This simple fit was developed to compare results from different cosmological simulations by making the assumption that most of the cosmology dependence can be encapsulated via the nonlinear mass scale, $M_\star$. Figure 7 shows our measurements and the fit for the c–M relation for $z = 0$ and 1; the result is given as a function of M_{200c} and not $M_{200c}/M_\star$ to enable easier comparison with the literature. (For the interested reader, comparisons of 12 different results from the literature with the above fit are given in Child et al. 2018, Figure A14.) We use the publicly available COLOSSUS package developed by Diemer (2018) to determine the fit over the required mass ranges. Overall, the fit captures our results well for both redshifts. The error bars for bin i are determined following the approach described in Bhattacharya et al. (2013, Equation (7)) by adding the error contribution from the individual concentration measurements and the Poisson error due to the finite number of halos in an individual bin in quadrature,

$$\Delta c_i(M) = \sqrt{\frac{\sum_j \delta c_j}{N_i} + \frac{c^2(M)}{N_i}}, \tag{19}$$

where Δc_i is the error for each bin and δc_j is the individual concentration measurement error for each halo. We have neglected the mass weighting that was applied in Bhattacharya et al. (2013, Equation (7)) after confirming that it only changes the result minimally. As is apparent in Figure 7, this error definition captures the larger uncertainty for the concentration measurement for small halos due to their less well-resolved profiles and for the larger halos due to their smaller number.

5.6. Infall Merger Tree Masses

In this section, we show a subset of the results from our merger trees by studying the infall halo mass functions (often referred to as "unevolved subhalo mass functions") and comparing to previous results. The infall halo mass function describes the mass distribution of halos just prior to merging with a more massive host. More information about merger tree statistics will be reported elsewhere.

We first calculate the first-order infall halo mass function (as defined in Jiang & van den Bosch 2014) of our halo merger trees corresponding to measuring the infall masses of all halos that were accreted along the most massive progenitor branch of a given merger tree. The infall halo mass function $dn/d\log(m/M)$ describes halos with infall mass m that were accreted onto any given tree with a final halo descendant mass M at $z = 0$. (All halo

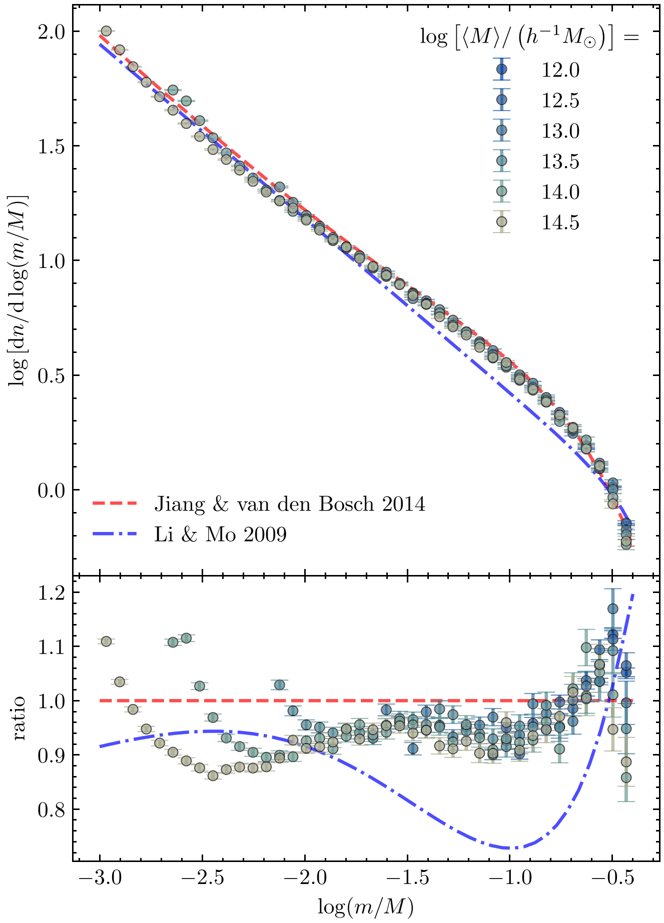

Figure 8. Upper panel: infall halo mass function at redshift $z = 0$ for first-order progenitors. Our results (points) are the averages for the six host halo mass bins specified in Table 2. An initial rise near the minimum halo mass of the simulation is not plotted for our host bins. Host halo and infall masses are determined using the FOF masses recorded from the Last Journey simulation. The fitting functions of Jiang & van den Bosch (2014; dashed line) and Li & Mo (2009; dashed–dotted line) for the unevolved subhalo mass function of first-order subhalos are also shown. Both functions fit the Millennium Simulation results of Li & Mo (2009), who used a virial mass definition for the halos. Lower panel: ratios of our mass functions and the Li & Mo (2009) fit to the Jiang & van den Bosch (2014) fitting function.

masses here are FOF masses with linking length $b = 0.168$, and the reported error bars are Poisson errors measured in each bin.) Note that these substructures are obtained from the core catalog using the host core tag described in Section 4.2, where only cores with host core tags equal to the final central tag at $z = 0$ were accreted by the main progenitor halos. The upper panel of Figure 8 shows the infall mass function for six $z = 0$ host halo mass bins. Table 2 lists the average host halo mass, mass range, and number of host halos in each bin.

We include a comparison with fitting functions derived in Jiang & van den Bosch (2014) and Li & Mo (2009) for the unevolved subhalo mass function of first-order subhalos in the Millennium Simulation. Jiang & van den Bosch (2014) noted that past work, including Li & Mo (2009), found the unevolved subhalo mass function of first-order subhalos to be approximately independent of host halo mass for a ΛCDM cosmology. This is consistent with the lack of mass dependence in our results, as shown in Figure 8. Our results are in qualitative agreement, lying in between the two fits. (We note that comparison is not fully direct because the mass ratios for the Last Journey results are for FOF masses, while the analytic fits use virial mass ratios.)

Table 2
Host Halo Mass Bins

$\log[\langle M\rangle/(h^{-1}M_\odot)]$	$M_{\rm min}$ $(10^{12}h^{-1}M_\odot)$	$M_{\rm max}$ $(10^{12}h^{-1}M_\odot)$	Host Halo Count
12.0	1.000	1.000	378,420
12.5	3.163	3.163	42,640
13.0	10.000	10.005	14,303
13.5	31.625	31.679	10,102
14.0	100.000	100.674	10,018
14.5	316.229	328.172	10,001

In addition, we calculate the "all-order" infall halo mass function at $z = 0$ of our halo merger trees. This is the mass function of all halos that have merged into either a $z = 0$ host halo or any of its progenitors. The upper panel of Figure 9 shows the infall mass function for the same six host halo mass bins used in Figure 8 (see Table 2). We include a comparison with the fitting function of Li & Mo (2009) for the unevolved subhalo mass function of all-order subhalos in the Millennium Simulation. The small difference in the amplitude is most likely due to the different host halo mass definition used in Li & Mo (2009) and the different cosmology employed. A more detailed investigation is forthcoming (I. Sultan et al., in preparation).

6. Data Release

As part of this paper, we make a subset of the simulation data from the Last Journey publicly available. We use the web-based data portal introduced in Heitmann et al. (2019a) for this purpose.[13] The data portal is based on Petrel,[14] a pilot project for data management and sharing, hosted at the ALCF. The web portal allows easy browsing of the available data products. Once the data sets of interest have been identified, Globus[15] enables fast and secure data transfer from Petrel to the target location. The data are stored in a customized HACC format, genericIO. This format was developed as an optimized read-and-write functionality for HACC at scale. The data portal provides a link to the genericIO repository,[16] which includes an example Python module for accessing the data.

The size of the full data set (close to 1 PB) by far exceeds the available storage for our project on Petrel; therefore, we carefully chose a subset of the data for the public release. The data products we make available—halo catalogs, halo particles, and a subset of the full particle data set at nine redshifts—enable the creation of synthetic sky catalogs targeted at cosmological survey science. For example, very similar data products from the Outer Rim simulation were used by the eBOSS collaboration to build HOD-based catalogs to enable clustering measurement of quasars via the correlation function (Zarrouk et al. 2018) and the power spectrum (Gil-Marín et al. 2018) and by the anisotropic clustering of quasars in configuration space (Hou et al. 2018). The downsampled particles allow the measurement of correlation functions and the generation of density and potential fields for a range of studies.

[13] https://cosmology.alcf.anl.gov
[14] https://press3.mcs.anl.gov/petrel
[15] https://www.globus.org
[16] https://xgitlab.cels.anl.gov/hacc/genericio.git

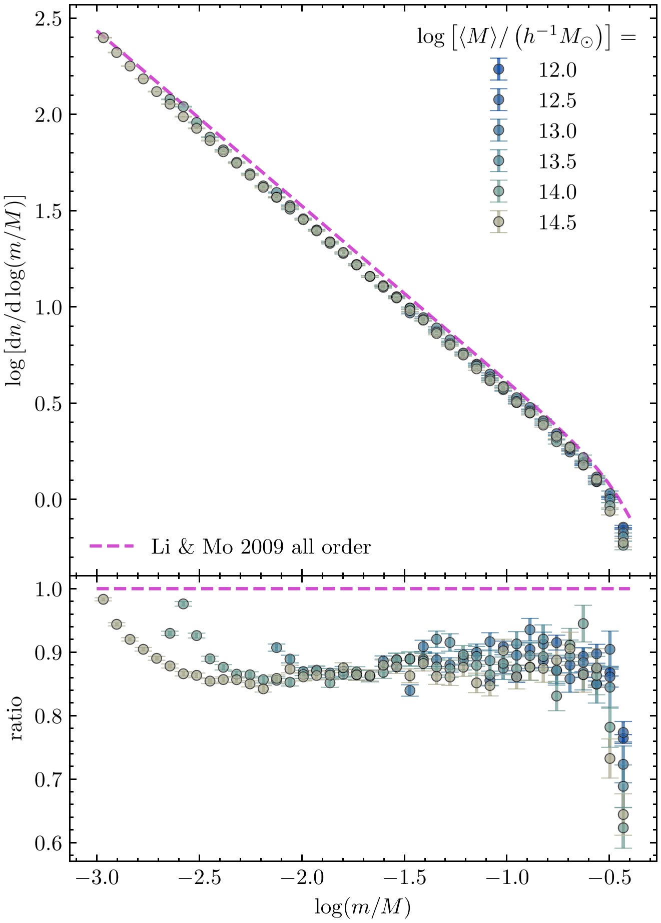

Figure 9. Upper panel: infall halo mass function at redshift $z = 0$ for all-order progenitors. Our results (points) are the averages for the six host halo mass bins specified in Table 2. An initial rise near the minimum halo mass of the simulation is not plotted for our host bins. The fitting function of Li & Mo (2009) for the unevolved subhalo mass function of all-order subhalos (dashed line) is also shown. As in Figure 8, the Last Journey results employ FOF masses, while the fitting function is based on virial masses. Lower panel: ratio of our mass functions to the Li & Mo (2009) fitting function. Note that the fit was derived for a cosmology with a higher value for $\sigma_8 = 0.9$ and a different mass definition for the host halo, likely accounting for the difference in the amplitude.

More specifically, we provide data products for the following nine redshifts:

$$z = \{1.49, 1.43, 0.86, 0.78, 0.54, 0.50, 0.21, 0.05, 0.0\}. \tag{20}$$

We enable access to the downsampled particle snapshots that contain 1% of the full particle outputs, randomly selected. In addition, we provide the results from the halo finder, including a range of halo properties. We release results from the FOF $b = 0.168$ finder at the same nine redshifts listed above. In particular, we provide a halo tag (which is reassigned at each snapshot and hence cannot be used to track halos over time), $m_{\rm halo}$ [$h^{-1}M_\odot$]; a halo center (x, y, z) [h^{-1} Mpc], measured at the halo's potential minimum; a halo center of mass (x, y, z) [h^{-1} Mpc]; and a mean velocity of the particles within the halo (v_x, v_y, v_z) [km s^{-1}]. The third data product we release is 1% of the particles that reside in halos. For all halos with less than 500 particles, we provide information for five halo particles. For each particle, we provide its position and velocity, a particle tag (which is the same at each snapshot), and the halo tag to connect the particles back to their host halo. All velocities for particles and halos are measured in comoving peculiar units. We summarize the released data products in Table 3.

Table 3
Released Data Products

Data Product	File Content	Overall Size
Downsampled particles, 1%	(x, y, z), (v_x, v_y, v_z), particle tag	3 TB
Halo particles, 1%, minimum five per halo	(x, y, z), (v_x, v_y, v_z), particle tag, halo tag	2.4 TB
FOF properties	halo tag, $m_{\rm halo}$, $(x, y, z)_{\rm pot}$, $(x, y, z)_{\rm COM}$, $(v_x, v_y, v_z)_{\rm COM}$	622 GB

Note. All data products are released at the following redshifts: $z = \{1.49, 1.43, 0.86, 0.78, 0.54, 0.50, 0.21, 0.05, 0.0\}$.

7. Summary and Outlook

Large-scale simulations have started to play a very important role in survey science over the last few years. Supercomputers have reached the level where simulations can cover enough volume and have sufficient resolution to enable the creation of realistic sky maps that can be used by cosmological surveys to investigate the correctness of their pipelines; model, investigate, and understand systematic effects; and carry out meaningful blind data challenges.

In this paper, we have described the Last Journey simulation, an extreme-scale gravity-only run, carried out on the Mira supercomputer. We evolved more than 1.2 trillion particles in a (3.4 h^{-1} Mpc)^3 volume using a cosmological model with parameter choices consistent with the analysis reported by the Planck Collaboration et al. (2020). We presented results for standard measurements of the simulation, including power spectra and mass functions, and compared them against results from emulators, finding very good agreement.

As cosmological simulations continue to grow in size, resource management for data analysis has become a key concern. There is an increasing emphasis on carrying out in situ analyses as the code is running, with the aim of minimizing I/O and offline computing and storage. The Last Journey run provides a good example of this imperative, as cosmological simulations prepare for the coming exascale era.

This paper describes the data sets available from the Last Journey run. These include particle light cones and detailed halo properties starting at redshift $z \approx 10$. Merger trees and core catalogs have also been produced. The use of core tracking integrated with merger trees provides a new method to follow the evolution of substructures within halos in very large simulations. Selected outputs from the simulations are publicly available at the HACC Simulation Data Portal.[17]

The simulation is designed to support a wide range of cosmological surveys across different wave bands. An important next step is the creation of a synthetic galaxy catalog using a semianalytic approach combined with empirical relationships to provide good fits to a host of observations. The core catalog will enable the tracking of substructures, serving as a replacement for

[17] https://cosmology.alcf.anl.gov

the commonly used subhalo merger tree and allowing for more accurate placements of model galaxies within the halos in the simulations. The core-tracking approach, in combination with the availability of particle light cones, also facilitates accurate small-scale lensing measurements, while the presence of tidal field and halo shape information will allow incorporation of IA modeling into such catalogs, adding realistic complexity and enabling testing of mitigation techniques. The presence of halo light-cone outputs out to high redshifts of $z \approx 10$ will enable improved modeling of the cosmic infrared background in support of next-generation CMB observations. The simulation outputs are also well suited for the creation of galaxy–galaxy and cluster weak lensing measurements, given the availability of full particle light cones of $\sim 10^9$ mass resolution at $z < 3$. Additional outputs of downsampled particle light cones at higher redshifts ($3 < z < 10$) will be used for creating CMB lensing maps.

The current paper is the first in a series of at least three Last Journey papers. The second paper will provide a detailed description of the core-tracking approach and the connected mass modeling method that is essential to enable the substitution of subhalo merger trees with core catalogs. The third paper will describe the results obtained from a SAM approach augmented with empirical modeling to build a detailed synthetic sky map.

We thank Malin Renneby for many early discussions regarding the project. We are grateful to Michael Bühlmann and Claude-André Faucher-Giguère for a careful reading of the manuscript and valuable comments. Argonne National Laboratory's work was supported under U.S. Department of Energy contract DE-AC02-06CH11357. An award of computer time was provided by the ASCR Leadership Computing Challenge (ALCC) program. This research was supported in part by DOE HEP's Computational HEP program. This research used resources of the Argonne Leadership Computing Facility at Argonne National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under contract No. DE-AC02-06CH11357. We are indebted to the ALCF team for their outstanding support and help in enabling us to carry out a full-scale simulation on Mira.

ORCID iDs

Katrin Heitmann https://orcid.org/0000-0003-1468-8232
Danila Korytov https://orcid.org/0000-0003-0801-8339
Eve Kovacs https://orcid.org/0000-0002-2545-1989

References

Abate, A., Aldering, G., Allan, S. W., et al. 2012, arXiv:1211.0310
Abell, P. A., Allison, J., Anderson, S. F., et al. 2009, arXiv:0912.0201 [astro-ph.IM]
Aghamousa, A., Aguilar, J., Ahlen, S., et al. 2016, arXiv:1611.00036 [astro-ph.IM]
Alam, S., de Mattia, A., Tamone, A., et al. 2020, arXiv:2007.09004 [astro-ph.CO]
Avila, S., Gonzalez-Perez, V., Mohammad, F. G., et al. 2020, MNRAS, 499, 5468
Behroozi, P. S., Wechsler, R. H., Wu, H.-Y., et al. 2013, ApJ, 763, 18
Benson, A. J. 2012, NewA, 17, 175
Bhattacharya, S., Habib, S., Heitmann, K., & Vikhlinin, A. 2013, ApJ, 766, 32
Bhattacharya, S., Heitmann, K., White, M., et al. 2011, ApJ, 732, 122
Blaizot, J., Wadadekar, Y., Guiderdoni, B., et al. 2005, MNRAS, 360, 159
Bleem, L. E., Bocquet, S., Stalder, B., et al. 2020, ApJS, 247, 25
Bocquet, S., Heitmann, K., Habib, S., et al. 2020, ApJ, 901, 5
Challinor, A., Chon, G., Colombi, S., et al. 2011, PolSpice: Spatially Inhomogeneous Correlation Estimator for Temperature and Polarisation, Astrophysics Source Code Library, ascl:1109.005
Child, H. L., Habib, S., Heitmann, K., et al. 2018, ApJ, 859, 55
Cohn, J. D. 2006, NewA, 11, 226
Crocce, M., Fosalba, P., Castander, F. J., & Gaztanaga, E. 2010, MNRAS, 403, 1353
Davis, M., Efstathiou, G., Frenk, C. S., & White, S. D. M. 1985, ApJ, 292, 371
Diemer, B. 2018, ApJS, 239, 35
Doré, O., Bock, J., Ashby, M., et al. 2014, arXiv:1412.4872 [astro-ph.CO]
Doré, O., Hirata, C., Wang, Y., et al. 2019, arXiv:1904.01174 [astro-ph.CO]
Fakhouri, O., & Ma, C.-P. 2008, MNRAS, 386, 577
Fang, X., Eifler, T., & Krause, E. 2020, MNRAS, 497, 2699
Feldman, H. A., Kaiser, N., & Peacock, J. A. 1994, ApJ, 426, 23
Gao, L., Navarro, J. F., Frenk, C. S., et al. 2012, MNRAS, 425, 2169
Gil-Marín, H., Guy, J., Zarrouk, P., et al. 2018, MNRAS, 477, 1604
Górski, K. M., Hivon, E., Banday, A. J., et al. 2005, ApJ, 622, 759
Habib, S., Morozov, V., Finkel, H., et al. 2012, Proc. SC12, arXiv:1211.4864
Habib, S., Pope, A., Finkel, H., et al. 2016, NewA, 42, 49
Han, J., Cole, S., Frenk, C. S., Benitez-Llambay, A., & Helly, J. 2018, MNRAS, 474, 604
Heinrich, J. 2003, Coverage of Error Bars for Poisson Data, CDF Note 6438, Batavia, IL: Fermilab
Heitmann, K., Bingham, D., Lawrence, E., et al. 2016, ApJ, 820, 108
Heitmann, K., Finkel, H., Pope, A., et al. 2019b, ApJS, 245, 16
Heitmann, K., Frontiere, N., Sewell, C., et al. 2015, ApJS, 219, 34
Heitmann, K., Lawrence, E., Kwan, J., Habib, S., & Higdon, D. 2014, ApJ, 780, 111
Heitmann, K., Lukić, Z., Fasel, P., et al. 2008, CS&D, 1, 015003
Heitmann, K., Ricker, P. M., Warren, M. S., & Habib, S. 2005, ApJS, 160, 28
Heitmann, K., Uram, T., Finkel, H., et al. 2019a, ApJS, 244, 17
Heitmann, K., White, M., Wagner, C., Habib, S., & Higdon, D. 2010, ApJ, 715, 104
Henriques, B., White, S., & Thomas, P. 2015, MNRAS, 451, 2663
Hong, S. E., Park, C., & Kim, J. 2016, ApJ, 823, 103
Hou, J., Sanchez, A. G., Scoccimarro, R., et al. 2018, MNRAS, 480, 2521
Hsu, C.-H., Ahrens, J., & Heitmann, K. 2010, in 2010 IEEE Pacific Visualization Symp. (Piscataway, NJ: IEEE), 81
Jenkins, A., Frenk, C. S., White, S. D. M., et al. 2001, MNRAS, 321, 372
Jiang, F., & van den Bosch, F. C. 2014, MNRAS, 440, 193
Kaiser, N. 1984, ApJ, 284L, 9
Kerscher, M., Szapudi, I., & Szalay, A. S. 2000, ApJL, 535, L13
Kitzbichler, M. G., & White, S. D. M. 2007, MNRAS, 376, 2
Knebe, A., Knollmann, S. R., Muldrew, S. I., et al. 2011, MNRAS, 415, 2293
Korytov, D., Hearin, A., Kovacs, E., et al. 2019, ApJS, 245, 26
Laureijs, R., Amiaux, J., Arduini, S., et al. 2011, arXiv:1110.3193 [astro-ph.CO]
Lewis, A., Challinor, A., & Lasenby, A. 2000, ApJ, 538, 473
Li, Y. 2019, ascl soft, Astrophysics Source Code Library, ascl:1906.017
Li, Y., & Mo, H. 2009, arXiv:0908.0301 [astro-ph.CO]
Lukić, Z., Heitmann, K., Habib, S., Bashinsky, S., & Ricker, P. M. 2007, ApJ, 671, 1160
Martínez, V. J., & Saar, E. 2002, Statistics of the Galaxy Distribution (Boca Raton, FL: Chapman & Hall/CRC)
Merson, A. I., Baugh, C. M., Helly, J. C., et al. 2013, MNRAS, 429, 556
Nagai, D., & Kravtsov, A. V. 2005, ApJ, 618, 557
Navarro, J. F., Frenk, C. S., & White, S. D. M. 1996, ApJ, 462, 563
Navarro, J. F., Frenk, C. S., & White, S. D. M. 1997, ApJ, 490, 493
Onions, J., Knebe, A., & Pearce, F. R. 2012, MNRAS, 423, 1200
Planck Collaboration, Aghanim, N., Akrami, Y., et al. 2020, A&A, 641, A6
Potter, D., Stadel, J., & Teyssier, R. 2017, ComAC, 4, 2
Rangel, E., Li, N., Habib, S., et al. 2016, in IEEE Int. Conf. on Cluster Computing (CLUSTERS) (Piscataway, NJ: IEEE), 30
Reid, B., & White, M. 2011, MNRAS, 417, 1913
Rodriguez-Gomez, V., Genel, S., Vogelsberger, M., et al. 2015, MNRAS, 449, 49
Rossi, G., Choi, P., Moon, J., et al. 2020, arXiv:2007.09002 [astro-ph.CO]
Sánchez, A. G., Baugh, C. M., & Angulo, R. E. 2008, MNRAS, 390, 1470
Schneider, A., Teyssier, R., Potter, D., et al. 2016, JCAP, 1604, 047
Smith, A., Burtin, E., Hou, J., et al. 2020, MNRAS, 499, 269
Smith, R. E., Scoccimarro, R., & Sheth, R. K. 2008, PhRvD, 77, 043525
Springel, V. 2005, MNRAS, 364, 1105
Srisawat, C., Knebe, A., Pearce, F. R., et al. 2013, MNRAS, 436, 150
Warren, M. S., Abazajian, K., Holz, D., & Teodoro, L. 2006, ApJ, 646, 881
Wechsler, R., & Tinker, J. 2018, ARA&A, 56, 435
Wetzel, A. R., Cohn, J. D., & White, M. 2009, MNRAS, 395, 1376
White, M., Blanton, M., Bolton, A., et al. 2011, ApJ, 728, 126
White, S. D. M., Davis, M., Efstathioui, G., & Frenk, C. S. 1987, Natur, 330, 451
Zarrouk, P., Burtin, E., Gil-Marín, H., et al. 2018, MNRAS, 477, 1639
Zel'dovich, Y. B. 1970, A&A, 5, 84